

# System Management

Edinburgh Release Update

# Edinburgh Release Task list

Current effort



- Refactor and improve the start/stop/restart capability of the SMA
  - Refactor to allow for more/better abstraction
  - Allow SMA to be Dockerized
- Go C SDK API work to be done
- Add CPU usage metrics (in addition to memory usage metrics)
  - Additional API (and pass through to the services) to be added to the SMA
  - Look at abstraction around metric APIs to allow for other implementations going forward
    - Example: allow memory and CPU metrics to be provided by external executable
- Add health/status check of the services to the SMA
  - This will be a call through to the configuration/registry service (Consul)
  - Allows a single point of entry for all EdgeX control plane needs
- **Stretch goal** – provide a translation layer (with abstraction) to offer SMA API via other protocol (like LWM2M, SNMP, etc.)

# Refactored start/stop/restart operations

- Requirements of system management start/stop functionality gathered @ last meeting
  - Given all the possible ways that a service like the SMA could start or start another service, we settled on support of the following:
    1. SMA would call Docker Compose to start and stop services
      - We choose not to support Docker calls directly because there are so many parameters that would have to be provided
      - Need proper abstraction to allow for other options going forward
        - Definitely want to keep REST calls to Docker in plan so as to decouple – but a bit complex at this time and may not find Docker API supportive
    2. SMA would call an executable (a binary) with a list of services it wants to start or stop.
      - Shell scripts could be handled by the author creating an executable that calls the appropriate script
      - No other parameters need to be provided as the executable will have its own config
      - Need proper abstraction to allow for other options going forward
- Determining success of the operation (start/stop) will be future feature
  - At least pass back the return code from call to Docker Compose or Executable
    - At least detect failure in a non-zero return
  - Desirable to have some sort of callback in future (stretch goal at best for Edinburgh)
  - Use consul to find status
  - Should at least provide log entry of operations for manual audit of what happened

# Start/Stop operation considerations

SMA **in** Docker container  
trying to start (or stop)  
EdgeX service **also in** a  
Docker container

Harder to do – but think  
docker-in-docker approach will  
work.

~~SMA **in** Docker container  
trying to start (or stop)  
EdgeX service that is **not in**  
a container~~

Doesn't seem to make sense.  
Any use cases?

~~SMA **not in** a Docker  
container trying to start  
(or stop) EdgeX service  
that is **in** a container~~

Doesn't seem to make sense.  
Any use cases?

SMA **not in** a Docker  
container trying to start  
(or stop) EdgeX service  
that is **not in** a container

Easy – SMA already does but  
we need to make sure  
interfaces are in place

Maybe some use cases for this in the future but agree to  
leave out for now

# Start/Stop operation POCs

device virtual test  
BB Tests

- POC A

- Test that docker-in-docker concept will allow docker-compose command from inside an image, will defer execution of docker-compose to outside the image (to the native OS)

✓ Completed

- POC B

- Create docker-in-docker image that contains Docker, Docker Compose and EdgeX docker-compose file
- Based on smallest image available (possibly dind to help provide Docker in Docker base)
- Create Dockerfile for this image as our starting point
- Be able to launch EdgeX with exec in and call to docker-compose up -f from inside this image

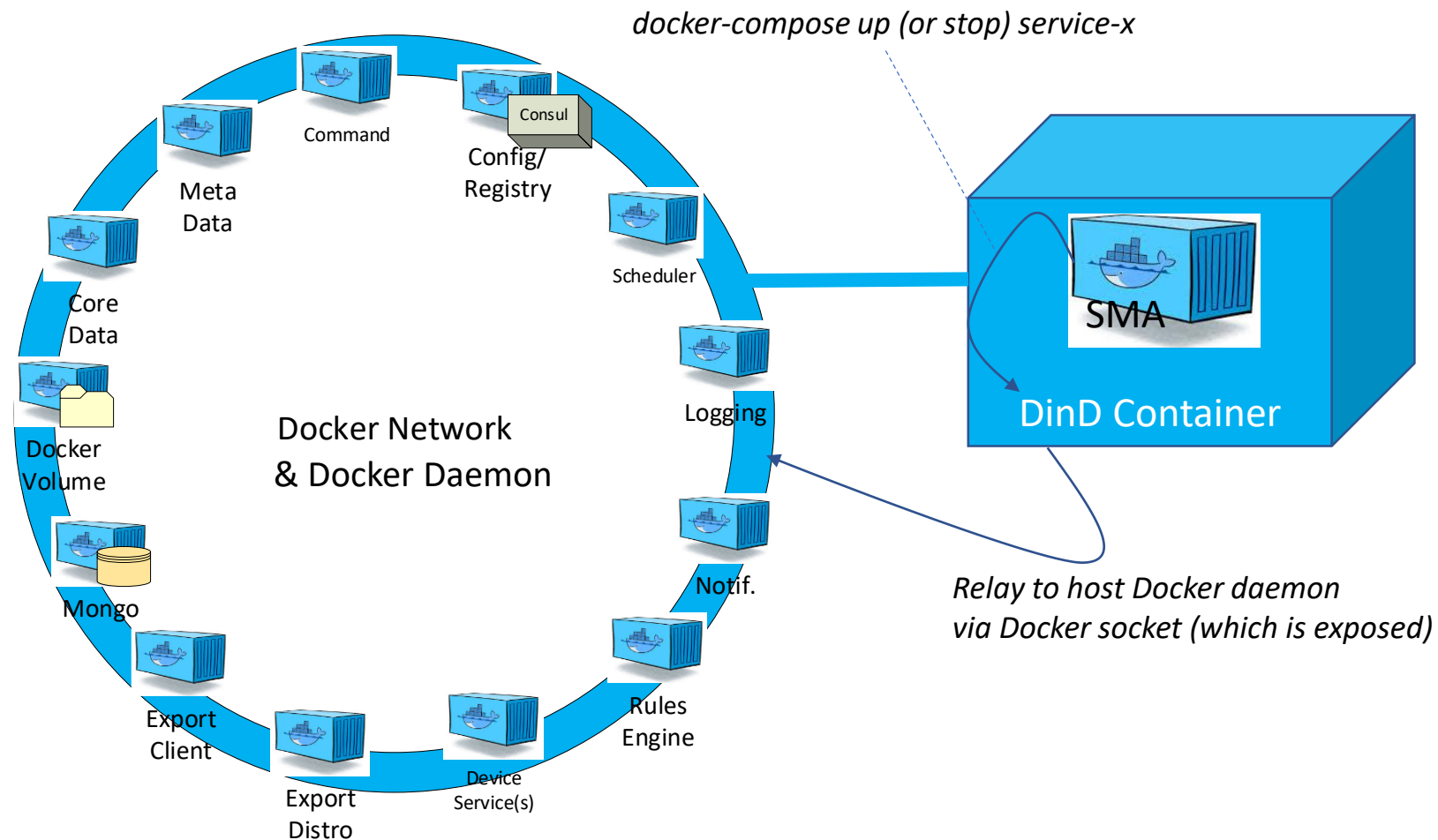
- POC C

- Add SMA to image above
- Be able to execute SMA REST call that triggers docker-compose call through docker-in-docker image to native OS to start/stop EdgeX

- Step D - redesign the APIs, interfaces and get community input on

- How SMA calls executable binary (and how it could support other options in the future)
- How SMA calls docker-compose to start/stop services (and how this could be used to support similar options in the future)

# System Management – Dockerized SMA



# System Management – Non-Dockerized SMA

